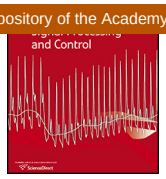
Biomedical Signal Processing and Control 62 (2020) 102041

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc



Assisted deep learning framework for multi-class skin lesion classification considering a binary classification support

Balazs Harangi*, Agnes Baran, Andras Hajdu

Faculty of Informatics, University of Debrecen, POB 400, 4002 Debrecen, Hungary

ARTICLE INFO

Article history:
Received 1 November 2019
Received in revised form 9 April 2020
Accepted 5 June 2020

Keywords:
Assisted learning
Deep learning
Ensemble learning
Skin lesion

ABSTRACT

In this paper, we propose a deep convolutional neural network framework to classify dermoscopy images into seven classes. With taking the advantage that these classes can be merged into two (healthy/diseased) ones we can train a part of the network regarding this binary task only. Then, the confidences regarding the binary classification are used to tune the multi-class confidence values provided by the other part of the network, since the binary task can be solved more accurately. For both the classification tasks we used GoogLeNet Inception-v3, however, any CNN architectures could be applied for these purposes. The whole network is trained in the usual way, and as our experimental results on the skin lesion image classification show, the accuracy of the multi-class problem has been remarkably raised (by 7% considering the balanced multi-class accuracy) via embedding the more reliable binary classification outcomes.

1. Introduction

Skin cancer is a common and locally destructive cancerous growth of the skin. It originates from the cells that line up along the membrane that separates the superficial layer of the skin from the deeper ones. As pigmented lesions occur on the surface of the skin, malignant behavior (e.g. melanoma) can be recognized early via visual inspection performed by a clinical expert. Dermoscopy is an imaging technique that eliminates the surface reflection of the skin. By removing surface reflection, more visual information can be obtained from the deeper levels of the skin.

In the last few years the computer aided diagnosis (CAD) is becoming more and more important in skin cancer detection [1]. As affordable mobile dermatoscopes are getting available to be attached to smart phones, the possibility for automated assessment is expected to positively influence corresponding patient care for a wide population. Given the widespread availability of high-resolution cameras, algorithms that can improve our ability to asses suspicions lesions can be of great value.

There is a long term history of computer aided dermoscopy image analysis and thus its literature is very verbose. The common protocol is to apply some pre-processing for image enhancement and artifact removal and then perform classification based on certain extracted features [2]. The current trend is to consider deep learning features as being superior over hand-crafted, or clinically inspired ones [1,3]. To extract deep learning features several convolutional neural network (CNN) systems have been considered. As for backbone CNN architectures the currently most popular ones are ResNet [4] and GoogLeNet [5].

Skin lesions can be categorized in numerous classes, however, the main practical task of the clinician is to differentiate between malignant and benign lesions, so the cardinal issue is to recognize their malignancy, that is, to be able to label them as benign/malignant (healthy/diseased or negative/positive) ones. Since benign/malignant appearances are usually sufficiently differ, the binary classification task can be solved with higher accuracy than the multi-class one in this field [6]. Moreover, if we can technically merge some classes into one, we have a chance to increase the number of training samples belonging to the two different classes without using any augmentation techniques which can also result in higher accuracy. Based on these observations, in this paper we propose a CNN architecture, which is simultaneously trained to solve a binary and a multi-class classification problem, where the two classes of the binary task represent the benign/malignant classes of the original 7-class skin lesion classification problem. Our methodological contribution is to incorporate the higher accuracy binary level classification confidence to support the final multi-class labeling. Naturally, our approach had to be realized in such a way that supports the efficient training of the CNN architecture in the common way via backpropagation. Our main motivation was to demonstrate the potential improvement of exploiting the higher classification accuracy on smaller num-

* Corresponding author.
E-mail address: harangi.balazs@inf.unideb.hu (B. Harangi).

https://doi.org/10.1016/j.bspc.2020.102041
1746-8094/

Table 1
Number of images corresponding to the 7 classes in the HAM10000 training set.

Classes	C_{NV}	C_{MEL}	C_{BCC}	C_{AKIE}	C_{BKL}	C_{DF}	C_{VASC}
Sample size	6705	1113	514	327	1099	115	142

ber of merged classes in a multi-class scenario and our empirical results have justified these expectations. In [7,8], a type of assisted learning was introduced for emotion recognition. The binary classification outcome there determined the further allowed labels in the multi-class problem, which is basically a classic decision tree-based method. Opposite to this approach we propose a network architecture that embeds binary classification in the training process. The connected convolutional neural networks learn together simultaneously and set their parameter to optimize the common loss function at the ensemble-system level based on the developed mathematical background.

The rest of the paper is organized as follows. In Section 2, we describe our novel methodology with presenting first a 7-class skin lesion dataset. We introduce our network architecture together with the proper formal description of applying binary classification support during training. Our experimental results are presented for the ISIC2018 challenge data in Section 3. We discuss on our hardware and training setups in Section 4 and also on how the proposed basic model can be improved further to increase its competitiveness in skin lesion classification. Finally, in Section 5 some conclusions are drawn.

2. Proposed methodology

2.1. Data

The organizers of the challenge International Skin Imaging Collaboration (ISIC) 2018: Skin Lesion Analysis Towards Melanoma Detection called for participation in developing efficient methods to classify skin lesion images into seven classes. Namely, the images are labeled according to the following classes of skin lesions:

- C_{BKL}: benign keratosis (solar lentigo/seborrheic keratosis/lichen planus-like keratosis),
- C_{DF}: dermatofibroma,
- C_{NV}: melanocytic nevus,
- C_{AKIE}: actinic keratosis or Bowen's disease,
- C_{BCC}: basal cell carcinoma,
- C_{MEL}: melanoma,
- C_{VASC}: vascular lesion.

Our data was extracted from the ISIC 2018: Skin Lesion Analysis Towards Melanoma Detection grand challenge datasets [9,10]. The ISIC2018 challenge has released an image set collected by Tschandl et al. [9] at the Department of Dermatology of the University of Vienna and at the Cliff Rosendahl in Queensland, Australia. The authors classified the collected images into seven generic classes because of simplicity and the reason that more than 95% of the lesions appearing in clinical practice fall into one of the seven diagnostic categories [9]. The published dataset contains 10,015 images for training, 193 for validation, and 1512 for testing purposes. The training set consists of images with manual annotations regarding the seven different classes in the following compounds: 6705 images with nevus lesions, 1113 with malignant skin tumors, 514 with basal cell carcinoma, 327 with actinic keratosis, 1099 with any benign keratosis, 115 dermatofibroma and 142 ones with vascular lesions. The number of images regarding these classes tries to follow the prevalence of the classes in the population, as well. Table 1 summarizes the number of images in the HAM10000 training-set

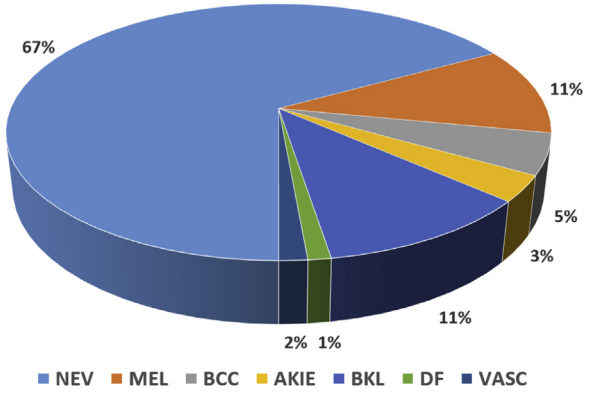


Fig. 1. The distribution of the images between the classes in the training set of HAM10000.

according to diagnosis and Fig. 1 depicts the distribution of the images among the classes.

The number of images in the certain classes (exclude the nevus one) is not sufficiently large for training deep CNNs. To increase the number of training images with also avoiding the over-fitting of the network and reducing the differences between the amount of images in the different classes, we have followed the commonly proposed solution [11] for the augmentation of the training dataset, such as cropping random samples from the images or horizontally flipping or rotating them at different angles. We also note that here the augmentation strategies to increase the size of the training dataset need to be applied after carefully understanding of the problem domain. It means that in order to avoid any modification of the characteristic texture of different type of lesions we could not use arbitrary scaling or aspect-ratio changes which are typically used. As the resolution of the images in the dataset are 450×600 pixels, but the applied CNN architectures originally require input image of spatial resolution 299×299, we randomly cropped sub-images with the required size from the original one instead of using scaling. Moreover, on these extracted images we applied rotating with randomly selected angle from the set $[90^\circ, 180^\circ, 270^\circ]$ and horizontal/vertical flipping. In order to create a more heterogeneous dataset we applied random brighten and contrast factors which are used to set the brightness and the contrast of the images randomly. Using these procedures, we have generated these modified training images and increased the original number of the sample images for the melanoma (4452), basal cell carcinoma (4626), actinic keratosis (4251), benign keratosis (4396), dermatofibroma (2415), and vascular lesion classes (2982).

The seven classes $C_1, \ldots, C_7$ of skin lesions can be further grouped as negative/positive (benign/malignant) ones as

$$\begin{aligned} C_{NEG} &= C_{BKL} \cup C_{DF} \cup C_{NV}, \\ C_{POS} &= C_{AKIE} \cup C_{BCC} \cup C_{MEL} \cup C_{VASC}. \end{aligned} \tag{1}$$

With this formulation we can set up a binary classification problem besides the original multi-class (7-class) one. Our motivation with adding the binary classification problem to the original one is that we can take advantage of the output of the binary classifier to make a finer labeling according to the 7 classes, since the simpler binary task (where the number of training samples corresponding to the different classes is larger) can be solved more accurately. For an impression of the characteristics of the lesions belonging to the seven classes, see also Fig. 2.

2.2. Network architecture

As for the design of our network architecture, our main motivation was to let the more reliable binary classifier to influence

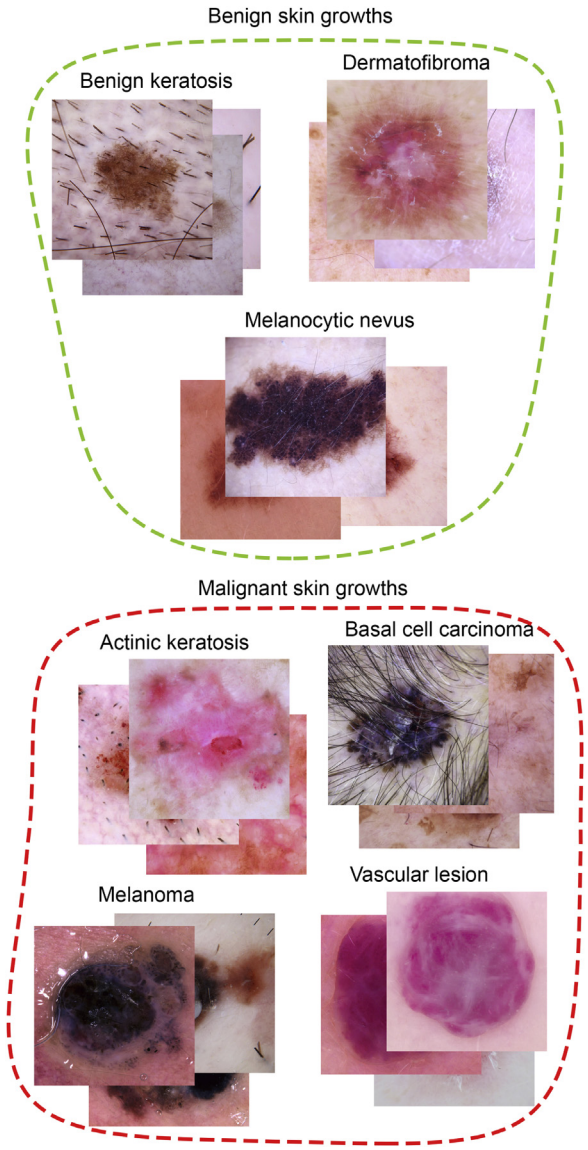


Fig. 2. Sample images for different types of skin lesions.

the output of the multi-level one. As backbone architectures for both the binary and multi-class classifiers we have considered GoogLeNet Inception-v3 [12], as GoogLeNet is reported to show a solid performance in skin lesion classification [3]. As common requirements of designing deep convolutional architectures, we also had to take care about to incorporate such functional elements that supports training via efficient backpropagation. Consequently, we had to involve a function, whose derivative can be given in closed form to describe the influence of the binary classifier.

To realize the above aims, we have considered the GoogLeNet Inception-v3 pre-trained model on ImageNet [13] and its layers have been fine-tuned in two times; once for the binary and once for the 7-class task. Then, these two fine-tuned models have composed into one network architecture as shown in Fig. 3. Namely, the proposed network can be divided into two main CNN branches with one of them is dedicated to binary, while the other for multi-class classification. Thus, the two branches result in respective 2D and 7D softmax layers, which are merged in a Support Training Layer (STL). Then, as the last layer of the network a 7D softmax layer is considered to address the original multi-class classification problem. At the STL layer, the probability values found by the binary classifier used to refine the corresponding multi-class probabilities via keeping/dropping them (CASE I) or via a simple multiplication (CASE II). More formally, let us suppose that at the STL layer we have class confidences p_{NEG} and p_{POS} with $p_{NEG}+p_{POS}=1$ by the binary, while $p_{BKL}+p_{DF}+p_{NV}+p_{AKIE}+p_{BCC}+p_{MEL}+p_{VASC}=1$ by the multi-class classifier regarding the corresponding classes. Moreover, let $[p]=\text{round}(p)$ denote the round operator, which provides 0 or 1 for both the binary/multi-class probabilities. Then, the confidence values for the final 7D softmax layer are calculated as:

- CASE I: $[p_{NEG}]*p_{BKL}$, $[p_{NEG}]*p_{DF}$, $[p_{NEG}]*p_{NV}$, $[p_{POS}]*p_{AKIE}$, $[p_{POS}]*p_{BCC}$, $[p_{POS}]*p_{MEL}$, $[p_{POS}]*p_{VASC}$,
- CASE II: $p_{NEG}*p_{BKL}$, $p_{NEG}*p_{DF}$, $p_{NEG}*p_{NV}$, $p_{POS}*p_{AKIE}$, $p_{POS}*p_{BCC}$, $p_{POS}*p_{MEL}$, $p_{POS}*p_{VASC}$

with a consequent normalization to sum them up to 1 in both cases.

To interpret better the differences between CASE I and II, notice that, with CASE I we basically follow a classic decision rule/tree model [7,8] with excluding the classes corresponding to the binary class having lower confidence; technically, the round operator provides a 0 multiplier accordingly. As a refined approach, CASE II follows a more dynamic way by tuning only the 7-class confidences with the binary ones. Though in our experiments CASE II has led to a remarkable improvement, we also enclose the performance of the CASE I approach as our initial attempt. In either case, for efficient computation we had to be able to embed our approach in the common training protocol by backpropagation, whose description is given next.

2.3. Training with backpropagation

In order to code the part of the backpropagation corresponding to the STL we have to compute the derivatives of the loss function with respect to the input data of this layer. For a formal description, let $\mathbf{x}=(x_1,\ldots,x_9)\in\mathbb{R}^9$ be the input of the STL layer, where $x_1=p_{NEG}$, $x_2=p_{POS}$ are the corresponding confidence results of the binary classifier, while the remaining 7 coordinates $x_3=p_{BKL}$, $x_4=p_{DF}$, $x_5=p_{NV}$, $x_6=p_{AKIE}$, $x_7=p_{BCC}$, $x_8=p_{MEL}$, $x_9=p_{VASC}$ of $\mathbf{x}$ describe the final probabilities provided by the 7-class CNN classifier branch of the network. When $\mathbf{z}\in\mathbb{R}^7$ is the output of the STL layer/the whole network, and L is the loss function, then we have to compute the derivatives $\frac{\partial L}{\partial x_i}$, $i=1,\ldots,9$. In CASE II during the forward pass the vector $\mathbf{z}$ is calculated as

$$\mathbf{z}=(x_1x_3,x_1x_4,x_1x_5,x_2x_6,x_2x_7,x_2x_8,x_2x_9)^T. \tag{2}$$

To calculate the required derivatives, we can use the chain rule:

$$\frac{\partial L}{\partial x_i}=\sum_{j=1}^{7}\frac{\partial L}{\partial z_j}\frac{\partial z_j}{\partial x_i}, \tag{3}$$

hence

$$\nabla_{\mathbf{x}}L=\begin{pmatrix} x_3 & x_4 & x_5 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & x_6 & x_7 & x_8 & x_9\\ x_1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & x_1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & x_1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & x_2 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & x_2 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & x_2 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & x_2 \end{pmatrix}\nabla_{\mathbf{z}}L, \tag{4}$$

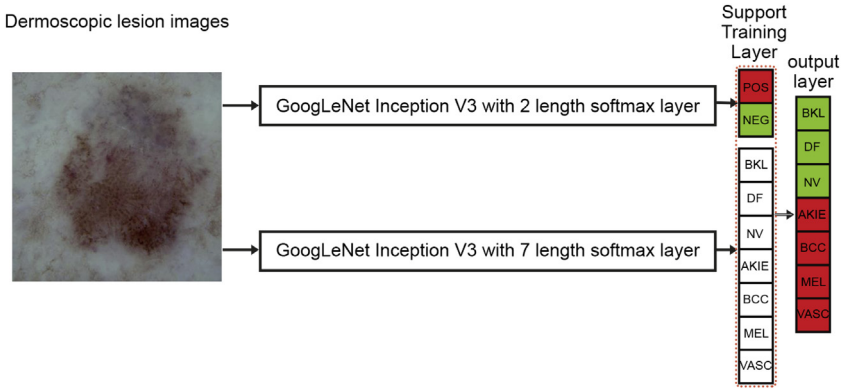


Fig. 3. The CNN architecture of our assisted training methodology.

where $\nabla_{\mathbf{x}} L$ and $\nabla_{\mathbf{z}} L$ denote the gradients of L with respect to $\mathbf{x}$ and $\mathbf{z}$.

In CASE I – using the previous notations – the vector $\mathbf{z}$ is given by

$$\mathbf{z} = ([x_1]x_3, [x_1]x_4, [x_1]x_5, [x_2]x_6, [x_2]x_7, [x_2]x_8, [x_2]x_9)^T. \quad (5)$$

Since the derivative of the round function is 0 everywhere but at 0.5, we obtain that the partial derivatives of L with respect to x_1 and x_2 will be equal to 0. It means that in this case the binary classifier as the part of the whole network cannot be trained. In other words, the binary classifier branch of the system will remain fixed during the whole training process, which lends a less flexible characteristics for CASE I. As a minor technical issue, notice that, the derivative of the round function does not exist at 0.5, however – similarly to the same phenomenon for ReLU –, this even occurs with 0 probability.

3. Experimental results

In this section, we summarize our experimental findings with a special motivation to see the binary classification assistance on the multi-class predictor for both CASE I and II. To be able to observe the improvement by this support, we give the quantitative results for the initial multi-class classifier without binary support, as well. We will start our presentation with this original setup.

The models are evaluated on the test set consisting of 1512 images provided by the ISIC 2018 challenge organizers. The evaluations have been performed on the official challenge web site according to the performance measures prescribed there. The submitted solutions are primarily ranked regarding the balanced multi-class accuracy (*BMA*) which is a commonly used measure in multi-class classification problems concerning imbalanced datasets. *BMA* is defined as the average sensitivity value obtained for the 7 classes. Moreover, as common performance measures like accuracy (*ACC*), sensitivity (*SE*), specificity (*SP*), positive predicted value (*PPV*) and area under the receiver operating characteristic curve (*AUC*) corresponding to each individual class have been calculated, as well.

For the sake of completeness, we enclose the detailed results of the models with presenting also their confusion matrices. Here, the diagonal elements represent the number of true positive cases for each class normalized by the cardinality of the given class (a.k.a. sensitivity), while the off-diagonal entries are to those elements that are mislabeled by the given classifier. Since the official challenge web site does not support this type of performance evaluation, we have considered the 20% of the augmented training set for validation (5964 images) to compute these confusion matrices. This is the reason why the sensitivity figures of the models regarding the different classes slightly differ in the tables and the corresponding confusion matrices. However, this is indeed a minor technical issue, since the trends naturally coincide.

Table 2
7-Class classification results without binary support.

	ACC	*SE*	*SP*	*PPV*	*AUC*
C_{BKL}	0.879	0.825	0.888	0.552	0.942
C_{DF}	0.985	0.545	0.998	0.889	0.937
C_{NV}	0.832	0.802	0.877	0.908	0.933
C_{AKIE}	0.973	0.163	0.997	0.583	0.908
C_{BCC}	0.966	0.731	0.981	0.716	0.981
C_{MEL}	0.874	0.573	0.912	0.454	0.877
C_{VASC}	0.988	0.571	0.998	0.870	0.994
BMA	0.602				

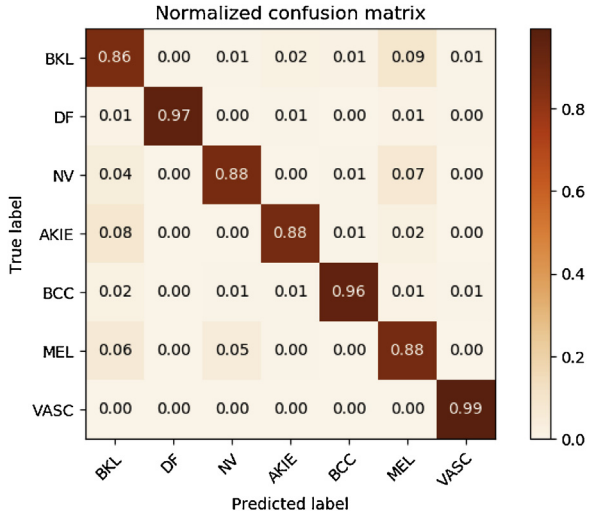


Fig. 4. Confusion matrix of 7-class classification results without binary support.

3.1. Results with no binary classification support

The simplest attempt to address this specific 7-class skin lesion classification task is to consider a single backbone CNN network with a final softmax layer of 7D. As for our current specific architecture it can be realized with restricting the network shown in Fig. 3 to its lower branch – a single GoogLeNet Inception-v3 with 7-class output – only. The *BMA* of this model has been found to be 0.602 with the additional performance measures are also shown in Table 2 for each class. To provide a more comprehensive analysis the proper corresponding confusion matrix is also given in Fig. 4.

Moreover, since simple binary (benign/malignant) classification can be highly informative for users of skin-related CAD systems,

Table 3
Binary classification results with merging the negative (benign) and positive (malignant) lesion classes.

	ACC	*SE*	*SP*	*PPV*	*AUC*
C_{NEG}	0.949	0.968	0.929	0.935	0.948
C_{POS}	0.949	0.929	0.968	0.965	0.948

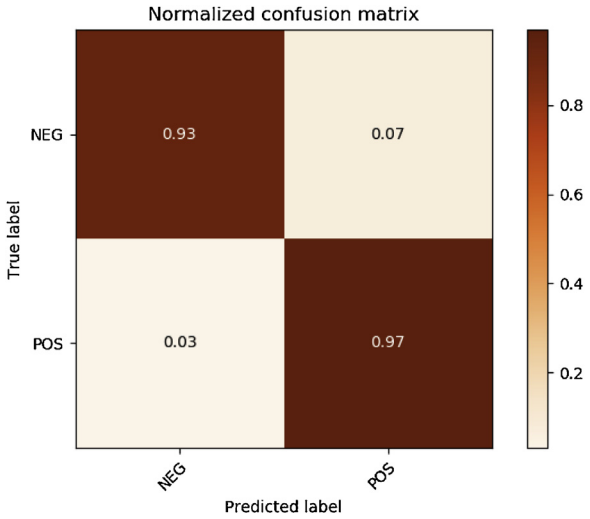


Fig. 5. Confusion matrix of the binary classification results.

in Table 3 we enclose the corresponding results. Notice that these binary classification results are derived from merging the 7 skin lesion classes according to (1) into a negative (healthy) and positive (diseased) class and considering the upper branch of our architecture from Fig. 3 – a single GoogLeNet Inception-v3 with binary output – for this task. For the sake of completeness, the confusion matrix for the simple binary classification setup is also given in Fig. 5.

The comparison of Tables 2 and 3 reflects our initiative to exploit the better binary classification performance in the multi-class problem.

3.2. Results with binary classification support

Now we turn to the exhibition of our experimental results corresponding to the main purpose of the current work with including binary classification support in the multi-class task. To be able to observe the continuous improvement in our approach, we present the experimental outcomes for both CASE I and II described in Section 2.2, that is, when binary support is involved in a rather drastic way (CASE I) and when it is applied to tune the multi-class predictions (CASE II).

As for CASE I, via multiplying the class probabilities of the 7 skin lesions either by 0 or 1 as the rounded class probabilities of the binary classifier, we follow a simple decision tree like model. In Table 4, we present the performance figures regarding the 7-class task solved by this approach according to the final softmax layer of the whole network. For more detailed comparative purposes, we enclose the corresponding confusion matrix in Fig. 6, as well.

Comparing Table 4 with Table 2 we can see that the CASE I approach has already outperformed the original single GoogLeNet Inception-v3 one regarding both its *BMA* = 0.639 value and the other measures. This result suggests that letting only the one-directional influence of the binary classifier to the multi-class one alone is already capable to lead to a slight improvement. However, notice that the derivative of the round function is zero everywhere (except 0.5), and during backpropagation the binary classifier branch cannot fine-tune its parameters. Moreover, in that case when the binary classifier misses the class label, the supporting model set each output neuron which belongs to one of the true class labels to zero.

Table 4
7-class classification results using binary support (CASE I).

	ACC	*SE*	*SP*	*PPV*	*AUC*
C_{BKL}	0.911	0.521	0.976	0.785	0.892
C_{DF}	0.981	0.409	0.999	0.900	0.865
C_{NV}	0.839	0.771	0.940	0.951	0.925
C_{AKIE}	0.977	0.372	0.995	0.696	0.773
C_{BCC}	0.966	0.688	0.984	0.744	0.928
C_{MEL}	0.882	0.602	0.917	0.481	0.889
C_{VASC}	0.991	0.657	0.999	0.820	0.960
BMA	0.639				

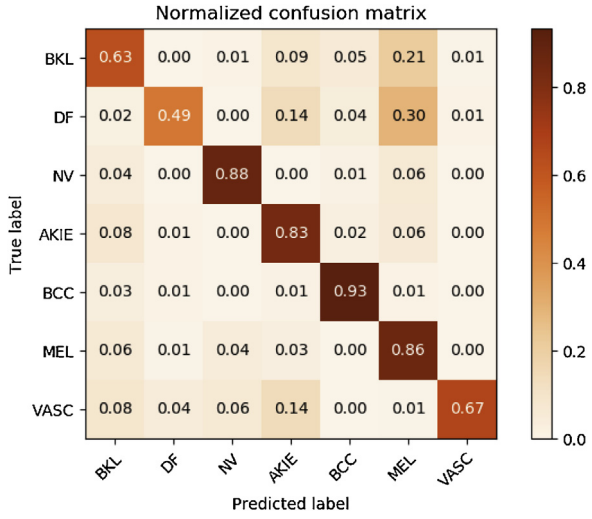


Fig. 6. Confusion matrix of the 7-class classification results for CASE I.

Table 5
7-Class classification results using binary support (CASE II).

	ACC	*SE*	*SP*	*PPV*	*AUC*
C_{BKL}	0.925	0.737	0.957	0.741	0.912
C_{DF}	0.983	0.477	0.999	0.913	0.867
C_{NV}	0.874	0.865	0.889	0.921	0.939
C_{AKIE}	0.976	0.326	0.995	0.636	0.827
C_{BCC}	0.971	0.570	0.997	0.930	0.938
C_{MEL}	0.913	0.450	0.972	0.675	0.812
C_{VASC}	0.989	0.571	0.999	0.909	0.924
BMA	0.677				

Next, we have applied the binary support with a more refined way according to CASE II with keeping the original probabilities of the binary classifier to tune the 7-class probabilities. The *BMA* of this model according the official ISIC 2018 challenge is 0.677. Similarly to the previous cases, we give the corresponding performance values and confusion matrix in Table 5 and Fig. 7, respectively.

Comparing Table 5 (CASE II) with Table 4 (CASE I), and Table 2 (no binary support), we can clearly observe that our initial purpose to improve multi-class classification performance by the binary one is finally successfully realized; overall, we could reach a 7% raise in balanced multi-class accuracy.

For the sake of completeness, we have made some comparative results regarding some state-of-the-art approaches which were used and evaluated during the ISIC 2018 Challenge: Skin Lesion Analysis Towards Melanoma Detection. Since there were many well known CNNs used as a proposed methodology for skin lesion classification, we have considered them as state-of-the-art solutions

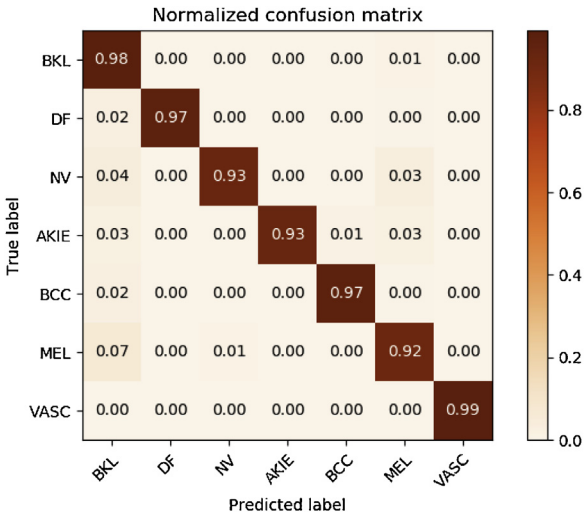


Fig. 7. Confusion matrix of the 7-class classification results for CASE II.

Table 6
Experimental classification results on the official ISIC 2018 test set.

Name of architecture	*BMA*
GoogLeNet Inception-v3 [12]	0.602
GoogLeNet-V3 with STL (CASE I)	0.639
GoogLeNet-V3 with STL (CASE II)	0.677
DenseNet-201 as trained by the Winner of Challenge [14]	0.868
DenseNet-201 trained only on ISIC 2018 dataset	0.712
DenseNet-201 trained only on ISIC 2018 dataset with STL (CASE II)	0.734
ResNet-101 [4]	0.675

and included in the quantitative comparison. Moreover, to show that our proposed method can be considered as a general framework which could increase classification accuracy when classes are mergeble in any way, we have involved other state-of-the-art CNN architectures into our final evaluation. Namely, some participants used the pre-trained ResNet-101 [4] and DenseNet-201 [14] in their solution. Since by DenseNet-201 superior results were achieved, we have also inserted this model in our framework to combine the seven-classes classifier with the binary classifier version using the proposed support training layer. After training DenseNet-201 on the ISIC 2018 Challenge dataset, we have evaluated its performance both with and without applying the binary classifier support using the ISIC Live 2018.3: Lesion Diagnosis automated evaluation system. The respective performances are also included in the quantitative comparison (see Table 6). As it can be seen from Table 6, the proposed method has remarkably improved the final classification accuracy for all the investigated CNN architectures including DenseNet-201, as well.

4. Discussion

As for the hardware environment, training has been performed on a computer equipped with an NVIDIA TITAN X GPU card with 7 TFlops of single precision, 336.5 GB/s of memory bandwidth, 3072 CUDA cores, and 12 GB memory. The convolutional filters of the network were found by a stochastic gradient descent algorithm iterated through 21 epochs till the validation accuracy started to drop after a while as depicted in Fig. 8. The reason of this approach lies in the fact that because of the small size of the dataset, a variant of GoogLeNet Inception-v3 pre-trained on ImageNet [13] has been considered, and its layers have been fine-tuned only separately for the binary and 7-class tasks. Then, these pre-trained branches had been fed into our assisted training architecture. This procedure is the explanation why the learning curve starts at quite

Table 7
Classification results using only the 7-class branch of the trained architecture.

	ACC	*SE*	*SP*	*PPV*	*AUC*
C_{BKL}	0.917	0.820	0.933	0.672	0.951
C_{DF}	0.983	0.500	0.998	0.880	0.897
C_{NV}	0.880	0.907	0.839	0.895	0.943
C_{AKIE}	0.976	0.395	0.993	0.607	0.855
C_{BCC}	0.972	0.613	0.995	0.891	0.956
C_{MEL}	0.903	0.579	0.944	0.569	0.864
C_{VASC}	0.991	0.657	0.999	0.920	0.996
BMA	0.643				

a high accuracy in Fig. 8, then drops a bit before the network learns the supporting phenomenon. Mini-batch size has been adjusted to 100, while the learning rate to 0.0001; other learning rates has been also tested, but iteration finished earlier with lower accuracy. As loss function we have selected cross-entropy as a common one for multi-class problems.

Our framework to support finer classification with a rawer one could be further tuned for the specific task. Hyperparameter optimization is one possibility to increase the accuracy, which – beyond the classic CNN parameters (stride, padding, number of layers/filters, dropout level, etc.) – may include a transformation of the simple approach to multiply the multi-class confidences with the corresponding binary ones. Moreover, ensemble-based systems are regularly reported to raise classification accuracy (see e.g. [15] for the same field). In our model, this approach could be realized with including more – either binary or multi-class – CNN components and select aggregation rules according to their outcomes and the appropriate way of providing the assistance by the binary classifier(s).

Since the original task was a 7-class classification one, we have not focused on the possible improvement of the binary classification outcome emerging the corresponding classes as described in Section 2.1. However, notice that it is possible to revert the direction of the support and to let the 7-class branch to assist the binary one. As for technical realization this could be reached by multiplying the p_{NEG} and p_{POS} probabilities with the normalized corresponding 7-class ones, so backpropagation can take place.

For the sake of an exhaustive analysis, we have also extracted the 7-class branch from the STL layer; the corresponding performance values are enclosed in Table 7. With this analysis we could check (see Tables 2 and 7) whether the 7-class branch was indeed able to improve during the assisted training as a standalone classifier. On the other hand, the whole architecture naturally has outperformed its 7-class branch regarding the classification task (see Tables 5 and 7).

5. Conclusion

In this paper, we have proposed a deep convolutional neural network architecture that supports multi-class classification by including the more reliable binary classification outcome in the final class probabilities. To realize this idea we have trained the same CNN architecture (GoogLeNet Inception-v3) for both a binary and multi-class task simultaneously with merging their softmax outputs on a support training layer with multiplying the multi-class confidences with the corresponding binary ones. In this way, we have achieved a remarkable improvement in a 7-class classification problem regarding skin lesions.

There is a natural limitation for our approach, namely when the classes cannot be merged directly to a smaller number of those. However, this issue can be addressed with applying some non-supervised technique like k-means clustering. This approach can lead to further generalizations with assigning dedicated branches in our ensemble for each recommended number of clusters deter-

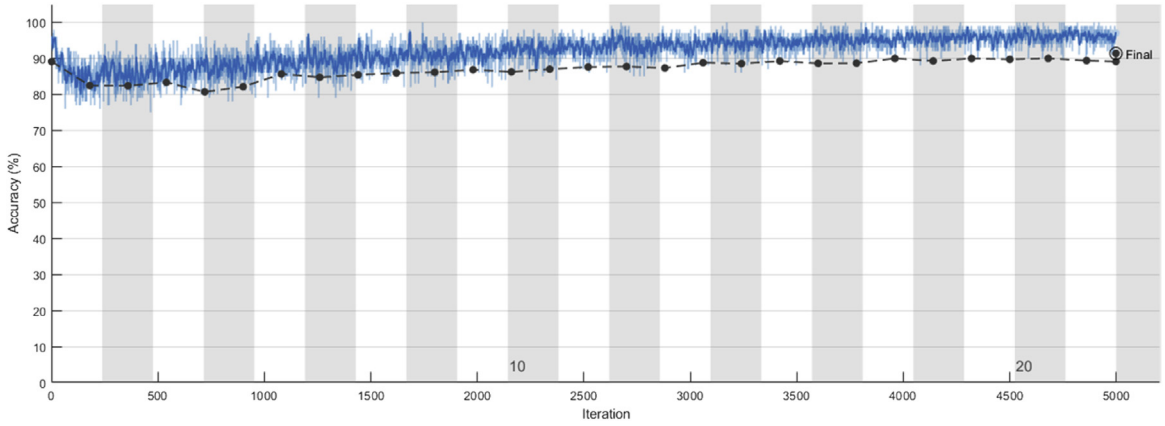


Fig. 8. Learning curve regarding training the 7-class classifier with a binary support (CASE II).

mined e.g. by the elbow method in k-means. Then, we can optimize assisted learning for that number of clusters corresponding to the specific task. Such ensembles consisting of several branches of CNNs can be optimized further with including a penalization term in the loss function for coinciding labeling to make the members more diverse besides keeping up overall classification accuracy.

Author contributions

Balazs Harangi: Conception and design of study, drafting the manuscript, approval of the final version of the manuscript. Agnes Baran: Conception and design of study, drafting the manuscript, approval of the final version of the manuscript. Andras Hajdu: Conception and design of study, revising the manuscript critically for important intellectual content, approval of the final version of the manuscript.

Acknowledgement

Research was supported in part by the Janos Bolyai Research Scholarship of the Hungarian Academy of Sciences and the projects EFOP-3.6.2-16-2017- 00015 supported by the European Union, co-financed by the European Social Fund.

References

[1] P. Tschandl, et al., Comparison of the accuracy of human readers versus machine-learning algorithms for pigmented skin lesion classification: an open, web-based, international, diagnostic study, Lancet Oncol. 20 (7) (2019) 938–947, http://dx.doi.org/10.1016/S1470-2045(19)30333-X.
[2] M.E. Celebi, T. Mendonca, J.S. Marques, Dermoscopy Image Analysis, CRC Press, 2018.
[3] A.C. Fidalgo Barata, E.M. Celebi, J. Marques, A survey of feature extraction in dermoscopy image analysis of skin cancer, IEEE J. Biomed. Health Inform. 23 (3) (2018) 1096–1109, http://dx.doi.org/10.1109/JBHI.2018.2845939.
[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 770–778, http://dx.doi.org/10.1109/CVPR.2016.90.
[5] C. Szegedy, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 1–9, http://dx.doi.org/10.1109/CVPR.2015.7298594.
[6] A. Esteva, B. Kuprel, R.A. Novoa, J. Ko, S.M. Swetter, H.M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, Nature 542 (2017) 115–118.
[7] X. He, W. Zhang, Emotion recognition by assisted learning with convolutional neural networks, Neurocomputing 291 (2018) 187–194, http://dx.doi.org/10.1016/j.neucom.2018.02.073.
[8] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106, http://dx.doi.org/10.1007/BF00116251.
[9] P. Tschandl, C. Rosendahl, H. Kittler, The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions, Sci. Data 5 (2018) 180161, http://dx.doi.org/10.1038/sdata.2018.161.
[10] N.C.F. Codella, D. Gutman, M.E. Celebi, B. Helba, M.A. Marchetti, S.W. Dusza, A. Kalloo, K. Liopyris, N.K. Mishra, H. Kittler, A. Halpern, Skin lesion analysis toward melanoma detection: a challenge at the 2017 international symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC), CoRR (2017), arXiv:1710.05006.
[11] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 60.
[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 2818–2826, http://dx.doi.org/10.1109/CVPR.2016.308.
[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, Int. J. Comput. Vision 115 (3) (2015) 211–252, http://dx.doi.org/10.1007/s11263-015-0816-y.
[14] G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely connected convolutional networks, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 2261–2269, http://dx.doi.org/10.1109/CVPR.2017.243.
[15] B. Harangi, Skin lesion classification with ensembles of deep convolutional neural networks, J. Biomed. Inform. 86 (2018) 25–32, http://dx.doi.org/10.1016/j.jbi.2018.08.006.